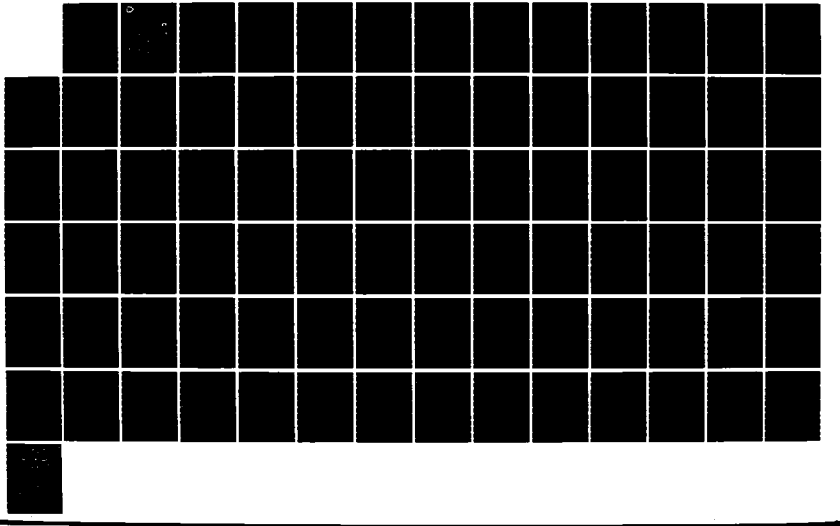
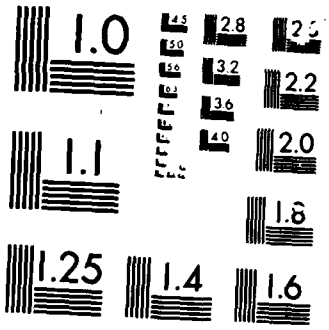
AD-A166 043 ADA (TRADE NAME) CURRICULUM BASIC ADA PROGRAMMING L202 1/1
TEACHER'S GUIDE LABORATORY MANUAL AND EXERCISES(U)
SOFTECH INC WALTHAM MA 1986 DAAB07-83-C-K514
UNCLASSIFIED F/G 9/2 NL




1.0
1.1
1.25
1.4
1.6
1.8
2.0
2.2
2.8
3.2
3.6
4.0
4.5
5.0
MICROCOPY RESOLUTION TEST CHART

1986

Ada® Training Curriculum

Basic Ada® Programming
L202
Teacher's Guide
Laboratory Manual and Exercises

Supersedes AD-A143583

U.S. Army Communications-Electronics Command
(CECOM)

Contract DAAB07-83-C-K506

Prepared By:

SOFTECH, INC.
460 Totten Pond Road
Waltham, MA 02154

BASIC Ada PROGRAMMING
(L202)
TEACHER'S GUIDE

LABORATORY MANUAL
AND EXERCISES

VG 956

INSTRUCTOR NOTES

THIS MANUAL CONTAINS EXERCISES TO BE COMPLETED DURING THE LABORATORY SESSIONS.

GOOD LUCK!

EXERCISES

INSTRUCTOR NOTES

ASSIGN EXERCISES 1 AND 2 AFTER SECTION 4, INTRODUCTION TO DATA.

YOU MAY HAVE TO PUT THE FOLLOWING PROCEDURE SHELL ON THE BOARD FOR SOME STUDENTS.

OTHERS WILL NOT REQUIRE THIS ADDITIONAL MATERIAL.

THIS IS STEP 1* → with Text_IO; use Text_IO;
procedure Conversion_Check is
--
-- declarations written here
--
THIS IS STEP 2* → package My_Int_IO is new Integer_IO (Integer);
use My_Int_IO;
begin -- Conversion_Check
--
-- executable code written here
--
end Conversion_Check;

THIS EXERCISE USES OBJECT DECLARATIONS, TYPE CONVERSION, AND I/O.

*THE STEPS REFER TO THE 2 STEPS REQUIRED FOR I/O PRESENTED IN SECTION 4.

EXERCISE 1

WRITE A PROCEDURE NAMED Conversion_Check TO DETERMINE HOW YOUR SYSTEM CONVERTS A VALUE SUCH AS 3.5 TO INTEGER.

INSTRUCTOR NOTES

THIS EXERCISE USES OBJECT DECLARATIONS AND ATTRIBUTES.

EXERCISE 2

WRITE A PROCEDURE CALLED Determine_Range_of_Integer TO DETERMINE THE SMALLEST AND LARGEST MACHINE REPRESENTABLE VALUES ON YOUR SYSTEM FOR THE PREDEFINED TYPE Integer.

INSTRUCTOR NOTES

ASSIGN EXERCISES 3, 4, 5, and 6 AFTER SECTION 5, ENUMERATION TYPES AND CONTROL STRUCTURES IS TAUGHT.

EXERCISE 3

```
with Text_IO; use Text_IO;
procedure Toll_Booth is

   type Vehicle_Type is ( ... );

   Toll_to_be_Paid : Float;

   procedure Calculate_Toll ( ... ) is
      ...

   end Calculate_Toll;

   package My_Float_Io is new Float_Io (Float);
   use My_Float_Io;

begin -- Toll_Booth;

   Calculate_Toll (Truck, 10, Toll_to_be_Paid);

   Put ("The toll to be paid is ");
   Put (Toll_to_be_Paid);

end Toll_Booth;
```

A. FINISH THE TYPE DECLARATION FOR type Vehicle_Type.

B. FINISH THE FORMAL PARAMETER DECLARATIONS FOR PROCEDURE Calculate_Toll.
HINT: SEE THE PROCEDURE CALL FOR THE ACTUAL PARAMETERS.

C. WRITE THE LOCAL PROCEDURE Calculate_Toll WHICH CALCULATES THE TOLL TO BE CHARGED A VEHICLE AT A TOLL BRIDGE IF THE TOLLS ARE 25 CENTS FOR CARS, 30 CENTS FOR CARS WITH TRAILERS, 15 CENTS FOR MOTORCYCLES, 50 CENTS FOR TRUCKS 10 TONS OR UNDER, AND 75 CENTS FOR TRUCKS OVER 10 TONS. FIRST CODE USING if STATEMENTS THEN USING case STATEMENTS.

INSTRUCTOR NOTES

ASSIGN AFTER L202 SECTION 5 IS TAUGHT.

THIS ASSIGNMENT IS REALLY MEANT TO BE USED WITH A LOOP.

THE STUDENTS WILL BE ASKED TO MODIFY THIS WITH LOOPS (EXERCISE 12) AND EXCEPTION HANDLERS (EXERCISE 29) LATER.

THE SOLUTION WHICH USES NEITHER A CASE NOR AN IF STATEMENT USES ATTRIBUTES.

EXERCISE 4

- WRITE A PROGRAM THAT WILL ACCEPT A LETTER FROM THE TERMINAL AND PRINT THE FULL COMMAND THAT THE LETTER ENTERED REPRESENTS.

POSSIBLE ENTRIES	SELECTED COMMAND
E	EDIT
L	LIST
H	HELP
P	OFFLINE PRINT
Q	QUIT

- WRITE FIRST USING AN if STATEMENT
- THEN WRITE USING A case STATEMENT
- THEN WRITE WITHOUT EITHER A case STATEMENT OR if STATEMENT
- YOU WILL NEED
  - Get
  - Put OR Put_Line

INSTRUCTOR NOTES

ASSIGN AFTER L202 SECTION 5 IS TAUGHT.

STUDENTS WILL BE ASKED TO MODIFY THIS WITH AN EXCEPTION HANDLER IN EXERCISE 28.

EXERCISE 5

WRITE A PROGRAM THAT ASKS THE USER WHAT THEIR NATIVE LANGUAGE IS AND THEN REPLIES WITH A WELCOME MESSAGE IN THAT LANGUAGE.

POSSIBLE NATIVE LANGUAGES AND WELCOME MESSAGES ARE:

SPANISH	-	SALUD, PESETAS Y AMOR.
FRENCH	-	BONJOUR, COMMENT C'EST VA?
ITALIAN	-	SALUTE! PIZZA, VINO E ALLEGRIA.
GERMAN	-	GRUESS GOTT! WIE GEHT'S?
ROMANIAN	-	SALUT! CE MAI FACI?
ENGLISH	-	AND NOW, HEEEEERE'S ADA!
VALLEYESE	-	LIKE TOTALLY AWESOME MORNING TO YA.

INSTRUCTOR NOTES

ASSIGN AFTER L202 SECTION 5 IS TAUGHT.

EXERCISE 6

A LIFE INSURANCE COMPANY BASES ITS PREMIUMS ON TWO CRITERIA, WEIGHT AND SMOKING HABITS. THE ACTIONS TAKEN BY THE COMPANY ARE SHOWN BELOW:

DOESN'T SMOKE NORMAL OR UNDER WEIGHT	DOESN'T SMOKE, OVERWEIGHT	SMOKES, NORMAL OR UNDER WEIGHT	SMOKES, OVER WEIGHT
INSURE USING REGULAR RATES	HOLD FOR FURTHER ANALYSIS	INSURE USING HIGHER RATES	DO NOT INSURE

WRITE A PROGRAM WHICH PROMPTS THE USER FOR TWO CRITERIA AND PRINTS TO THE TERMINAL THE ACTION TAKEN BY THE INSURANCE COMPANY. WRITE THE PROCEDURE WITH EITHER if STATEMENTS, case STATEMENTS, OR BOTH.

INSTRUCTOR NOTES

ASSIGN AFTER L202 SECTION 6 IS TAUGHT.

- THE NOTE IN A MEANS "DON'T BE ALARMED OR WORRIED ABOUT FORMATTING OUTPUT."
- THEY WILL NEED THE SQUARE_ROOT FUNCTION FOR THIS. BE SURE TO PASS OUT A COPY OR HAVE IT ONLINE FOR THEM TO COPY.
- IN EXERCISE 12, THEY WILL BE ASKED TO WRITE THE RESULTS TO A FILE.
- HERE IS ONE POSSIBLE IMPLEMENTATION OF A SQUARE ROOT ROUTINE:

```
function Sqrt (Number : Integer) return Float is

   Epsilon : constant := 0.000001;
   Root : Float;
   New_Number : Float;
begin -- Sqrt

   if Number = 0 then
         return 0.0;
   else
         New_Number := Float(Number);
         Root := 1.0;
         Root := (New_Number / Root + Root) / 2.0;

         while abs (New_Number / Root ** 2 - 1.0) > = Epsilon
            loop
               Root := (New_Number / Root + Root) / 2.0;
            end loop;

         return Root;
      end if;

end Sqrt;
```

EXERCISE 7

A. WRITE A PROGRAM WHICH PRINTS TO THE TERMINAL THE NUMBERS 1 THROUGH 5 AND THEIR CORRESPONDING SQUARE ROOTS.

NOTE: ALL THE OUTPUT WILL BE DIRECTED TO THE SAME LINE. ONLY WHEN THE END OF THE LINE IS REACHED WILL A NEW LINE BE STARTED.

b. MODIFY THE PROGRAM SO THAT THE OUTPUT IS IN TABULAR FORM.

EXAMPLE:

1	1.0
2	1.xxxxxx
3	1.xxxxxx
4	2.0
5	2.xxxxxx

INSTRUCTOR NOTES

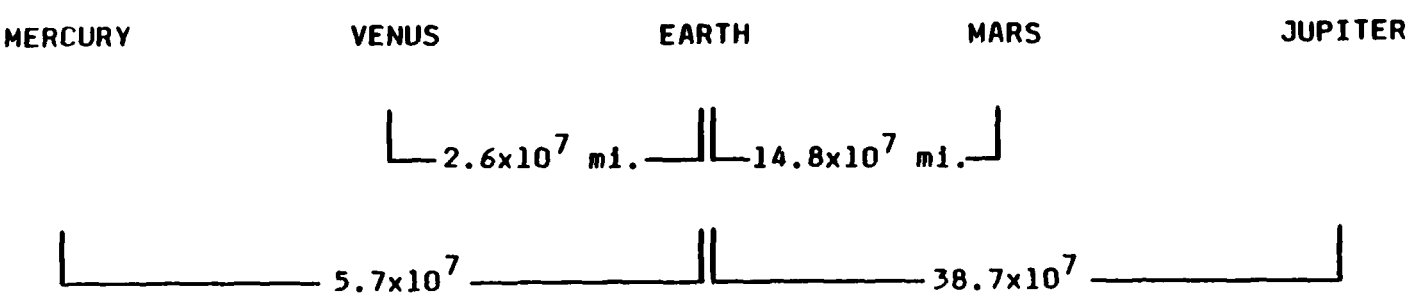
ASSIGN AFTER L202 SECTION 6 IS TAUGHT.

IF STUDENTS USE FIXED POINT ARITHMETIC, THE RESULT WILL NEED TO BE CONVERTED TO THE FIXED POINT TYPE.

EXERCISE 8

WRITE A PROGRAM THAT CALCULATES HOW LONG IT WOULD TAKE FOR THE SPACE SHUTTLE TO TRAVEL FROM EARTH TO ANY OF FOUR PLANETS. YOUR PROGRAM SHOULD ASK THE USER WHICH PLANET THE SHUTTLE IS GOING TO, AND THEN PRINT ON THE TERMINAL THE NUMBER OF DAYS IT WILL TAKE. ASSUME THAT THE SHUTTLE HAS A VELOCITY OF 3.6×10^5 MILES PER DAY. USE THE DISTANCES BELOW.

NOTE: YOU CAN USE FIXED POINT TYPE DECLARATIONS WITH A DELTA OF 1.0
OR FLOATING POINT TYPE DECLARATIONS.



INSTRUCTOR NOTES

ASSIGN AFTER L202 SECTION 6 IS TAUGHT. THIS COULD BE USED AS AN IN-CLASS ASSIGNMENT.

THIS COULD BE IMPLEMENTED AS A PROCEDURE WITH AN out PARAMETER OR AS A FUNCTION.

WHEN ASSIGNED FOR LAB, HAVE THE STUDENTS NEST THE SUBPROGRAM WITHIN AN OUTER PROGRAM WHICH ACCEPTS INPUT FROM THE TERMINAL.

THE PREFERRED SOLUTION USES SUBTYPES.

STUDENTS WILL BE ASKED TO MODIFY THEIR SOLUTION TO INCLUDE EXCEPTION HANDLERS IN EXERCISE 31.

EXERCISE 9

WRITE A PROGRAM THAT ACCEPTS AN INTEGER BETWEEN 100 AND 999, INCLUSIVE, CALCULATES THE SUM OF THE INDIVIDUAL DIGITS, AND RETURNS THE VALUE CALCULATED.

EXAMPLE:

361 ⟶ 10

202 ⟶ 4

980 ⟶ 17

HINT: USE rem OR mod.

ASSIGN AFTER L202 SECTION 7 IS TAUGHT.

ANSWER:

```
Char := Character'Val(Character'Pos(Char)+32);
```

EXERCISE 10

WRITE THE ALGORITHM NECESSARY TO CONVERT AN UPPER CASE CHARACTER TO A LOWER CASE CHARACTER.

HINT: Character'Pos ('a') - Character'Pos('A') = 32.

USE ATTRIBUTES. THIS CAN BE DONE WITH ONE LINE.

INSTRUCTOR NOTES

POINT OUT THAT IT IS IMPLEMENTATION DEFINED WHETHER OR NOT A FILE IS CLOSED WHEN THE USER FORGETS TO CLOSE A FILE EXPLICITLY.

I/O TUTORIAL -- USER FILES

THE PROGRAM BELOW ILLUSTRATES HOW TO DIRECT OUTPUT TO A FILE.

```
with Math_lib;
with Text_IO; use Text_IO;
procedure Compute_Sine is

   Sine    : Float;
   Results : File_Type;                                  -- creates file object

   package Flt_IO is new Float_IO (Float);

begin

   Create (Results, Out_File, "Results.Dat");            -- creates external file

   Sine := Math_Lib.Sin (.052360);

   Put (Results, "The sine of .052360 radian is ");      -- output to file

   Flt_IO.Put (Results, Sine);                           -- output to file

   Close (Results);

end Compute_Sine;                                        -- close file
```

INSTRUCTOR NOTES

I/O TUTORIAL -- USER FILES

- Create (Results, Out_File, "Results.Dat"); CREATES THE EXTERNAL (ACTUAL) FILE "Results.Dat" AND ASSOCIATES IT WITH THE INTERNAL FILE Results. (Results WAS DECLARED TO BE A FILE BY THE STATEMENT, Results : File_Type;.)

- THE THIRD PARAMETER, THE NAME OF THE EXTERNAL FILE, COULD HAVE BEEN OMITTED. THIS WOULD CAUSE A TEMPORARY, UNNAMED FILE TO BE CREATED AND ASSOCIATED WITH THE INTERNAL FILE Results.

- AN OPEN STATEMENT COULD HAVE BEEN USED IF THE FILE "Results.Dat" HAD ALREADY EXISTED.

```
Open (Results, Out_File, "Results.Dat");
```

  THE SECOND PARAMETER, Out_File, SPECIFIES THE MODE OF THE FILE. THE MODE CAN BE In_File (FOR INPUT ONLY FILES) AND Out_File (FOR OUTPUT ONLY FILE).

- THE CLOSE STATEMENT CLOSES THE FILE.

INSTRUCTOR NOTES

ASSIGN AFTER L202 SECTION 8 IS TAUGHT.

EXERCISE 11

WRITE A PROCEDURE WHICH OUTPUTS TO A FILE IN TABULAR FORM THE FACTORIALS FOR ALL THE NUMBERS BETWEEN TWO NUMBERS ENTERED BY THE TERMINAL, I.E. BETWEEN 1 AND 5 IF 1 AND 5 WERE ENTERED.

HINT: WRITE A LOCAL PROCEDURE WHICH SKIPS TO A NEW LINE BEFORE OUTPUTTING THE INFORMATION.

INSTRUCTOR NOTES

ASSIGN AFTER L202 SECTION 8 IS TAUGHT. MANY STUDENTS MAY HAVE DONE THIS ALREADY. ASSIGN TO THOSE WHO HAVEN'T. THEY WILL NEED THE MODIFIED VERSION LATER.

EXERCISE 12

A. MODIFY THE PROGRAM FROM EXERCISE 3 SO THAT IT LOOPS UNTIL "DONE" IS ENTERED.

B. MODIFY THE PROGRAM FROM EXERCISE 7 SO THAT THE OUTPUT IS PRINTED INTO A FILE.

INSTRUCTOR NOTES

ASSIGN AFTER SECTION 8 IS TAUGHT.

IF THE STUDENT IS CONFUSED BY CROSS PRODUCT, SUGGEST THEY JUST IMPLEMENT THE HINT OR GO ON TO ANOTHER PROBLEM. THE EXPLANATION IS NOT IN THE SCOPE OF THIS COURSE.

EXERCISE 13

WRITE A PROGRAM TO COMPUTE THE CROSS PRODUCT OF TWO VECTORS (X, Y, Z) AND (A, B, C) AND PRINT THE ANSWER IN THE FORM (v1, v2, v3).

HINT: IF (a1...a3) and (b1...b3) ARE VECTORS THEN

cross product = (a1...a3) * (b1...b3) = (a2b3-a3b2, a3b1-a1b3, a1b2-a2b1)

INSTRUCTOR NOTES

ASSIGN AFTER L202 SECTION 8 IS TAUGHT.

STUDENTS WILL BE ASKED TO MODIFY THIS IN EXERCISE 19.

EXERCISE 14

WRITE A PROGRAM TO MULTIPLY THE TWO MATRICES $\begin{bmatrix} 1 & 2 \\ -1 & 3 \end{bmatrix}$ and $\begin{bmatrix} -2 & 4 \\ 5 & -3 \end{bmatrix}$

AND PRINT THE RESULT TO THE SCREEN IN THE FORM $\begin{bmatrix} r1 & r2 \\ r3 & r4. \end{bmatrix}$

HINT : IF $\begin{bmatrix} X11 & X12 \\ X21 & X22 \end{bmatrix}$ AND $\begin{bmatrix} Y11 & Y12 \\ Y21 & Y22 \end{bmatrix}$ ARE MATRICES THEN

$$\begin{bmatrix} X11 & X12 \\ X21 & X22 \end{bmatrix} * \begin{bmatrix} Y11 & Y12 \\ Y21 & Y22 \end{bmatrix} = \begin{bmatrix} X11*Y11 + X12*Y21 & X11*Y12 + X12*Y22 \\ X21*Y11 + X22*Y21 & X21*Y12 + X22*Y22 \end{bmatrix}$$

INSTRUCTOR NOTES

ASSIGN AFTER L202 SECTION 8 IS TAUGHT.

STRESS THAT THEY ARE USING ARRAYS HERE. A LATER EXERCISE ASKS FOR COMPLEX NUMBERS TO BE IMPLEMENTED AS RECORDS (EXERCISE 20).

EXERCISE 15

IMPLEMENT THE COMPLEX NUMBER AS AN ARRAY:

A. WRITE A FUNCTION WHICH WILL ADD COMPLEX NUMBERS.

HINT: A COMPLEX NUMBER HAS THE FORM a+bi

THE SUM OF a+bi AND c+di IS

(a+c) + (b+d)i

B. WRITE A FUNCTION WHICH MULTIPLIES TWO COMPLEX NUMBERS.

HINT: a+bi * c+di is

(ac-bd) + (ad+cb)i

INSTRUCTOR NOTES

C. WRITE A PROCEDURE THAT ACCEPTS TWO COMPLEX NUMBERS AND OUTPUTS THE SUM AND THE PRODUCT.

HINT: WRITE A SEPARATE PROCEDURE TO FORMAT THE OUTPUT

D. WRITE A PROCEDURE WHICH ACCEPTS A COMPLEX NUMBER AND AN INTEGER EXPONENT AND OUTPUTS THE COMPLEX NUMBER RAISED TO THAT POWER.

HINT: REPEATED CALLS TO THE MULTIPLICATION FUNCTION.

REPRESENT THE COMPLEX NUMBERS AS AN ARRAY.

INSTRUCTOR NOTES

ASSIGN AFTER L202 SECTION 8 IS TAUGHT.

THE STUDENTS WILL BE ASKED TO WRITE A REPORT USING THESE RESULTS.

STRESS THIS EXERCISE. IT HAS PROVED VERY EDUCATIONAL IN THE PAST. HAVE STUDENTS PLACE PRIORITY ON THIS OVER OTHER ASSIGNMENTS.

EXERCISE 16

A FRUIT-GROWING COMPANY NAMED ELOD HAS LOCATIONS IN HAWAII, MEXICO AND THE VIRGIN ISLANDS. AT EACH LOCATION, THERE ARE THREE FRUITS GROWN, GRAPEFRUIT, LIMES, AND PINEAPPLES, AND TEN TREES OF EACH FRUIT. BELOW ARE LISTED THE ANNUAL YIELD IN BUSHELS OF EACH TREE.

TREE NUMBER:	1	2	3	4	5	6	7	8	9	10
HAWAII										
GRAPEFRUITS	3	6	9	8	5	2	0	1	9	5
PINEAPPLE	2	9	8	0	6	5	3	7	4	0
LIMES	0	1	2	0	0	3	5	0	0	1
MEXICO										
GRAPEFRUITS	2	5	3	1	8	9	0	0	2	2
PINEAPPLE	8	7	6	4	3	0	0	3	9	2
LIMES	1	9	8	9	7	6	4	6	6	7
VIRGIN ISLANDS										
GRAPEFRUITS	1	0	1	1	2	4	9	9	8	7
PINEAPPLES	0	0	3	4	9	4	3	2	4	5
LIMES	3	4	2	1	6	8	9	0	7	7

A. WRITE A PROCEDURE THAT CALCULATES THE TOTAL ANNUAL YIELD IN BUSHELS.

B. ADD A PROCEDURE THAT CALCULATES THE ANNUAL YIELD IN EACH SEPARATE LOCATION.

C. ADD A PROCEDURE THAT CALCULATES THE TOTAL ANNUAL YIELD OF EACH FRUIT.

D. CALCULATE THE NUMBER OF TREES WHOSE ANNUAL YIELD IS GREATER THAN 5 BUSHELS.

E. CALCULATE THE NUMBER OF GRAPEFRUIT TREES WHOSE ANNUAL YIELD IS 8 BUSHELS.

F. FOR EACH LOCATION, DETERMINE THE FRUIT CROPS WITH THE BEST AND WORST YIELD.

INSTRUCTOR NOTES

ASSIGN AFTER L202 SECTION 8 HAS BEEN TAUGHT.

EXERCISE 17

A. WRITE A PROGRAM THAT ALPHABETIZES A LIST OF TEN WORDS. YOUR PROGRAM SHOULD ASK THE USER TO TYPE IN ANY TEN SIX-LETTER WORDS. THE LIST SHOULD THEN BE PRINTED IN ALPHABETICAL ORDER.

B. MODIFY YOUR PROGRAM SO THAT IT CREATES ONE LONG STRING OF THE ALPHABETIZED LIST WITH COMMAS AND SPACES BETWEEN THE WORDS.

C. MODIFY YOUR PROGRAM SO THAT IT WILL ALSO PRINT SEPARATELY THE FIRST AND LAST CHARACTER OF THIS FINAL STRING.

D. MODIFY YOUR PROGRAM SO THAT IT ALSO PRINTS A MESSAGE INDICATING WHETHER ANY CHANGES HAD TO BE MADE TO THE ORDER OF THE ORIGINAL LIST OF WORDS.

E. CAN YOU DECLARE A STRING TYPE OF UNCONSTRAINED LENGTH?

INSTRUCTOR NOTES

ASSIGN AFTER L202 SECTION 8 IS TAUGHT.

EXERCISE 18

WRITE A PROGRAM THAT CALCULATES ANY ROW OF PASCAL'S TRIANGLE. THE PROGRAM SHOULD ASK THE USER FOR THE DESIRED ROW, AND PRINT THE RESULT.

PASCAL'S TRIANGLE LOOKS LIKE THIS:

ROW

```
1                1                    NOTE THAT EACH NUMBER IS THE SUM
2              1   1                  OF THE TWO ABOVE IT.
3            1   2   1
4          1   3   3   1
5        1   4   6   4   1
6      1   5  10  10   5   1
7    1   6  15  20  15   6   1
.                .
.                .
.                .
```

EXAMPLE. WHEN THE USER ENTERS 6, YOUR PROGRAM SHOULD OUTPUT

```
1 5 10 10 5 1
```

INSTRUCTOR NOTES

ASSIGN AFTER L202 SECTION 8 HAS BEEN TAUGHT.

EXERCISE 19

GENERALIZE THE PROGRAM WRITTEN IN EXERCISE 14 TO MULTIPLY ANY TWO 2X2 MATRICES.

INSTRUCTOR NOTES

ASSIGN AFTER L202 SECTION 9 IS TAUGHT.

FOR THOSE WHO IGNORED THE HINT AND ALREADY DID IT WITH RECORDS, HAVE THEM GO BACK AND REDO IT USING ARRAYS.

EXERCISE 20

REWRITE THE PROGRAMS FOR EXERCISE 15, THIS TIME REPRESENTING COMPLEX AS A RECORD.

INSTRUCTOR NOTES

ASSIGN AFTER L202 SECTION 9 IS TAUGHT.

EXERCISE 21

A LIQUOR STORE NEEDS AN INVENTORY SYSTEM.

A. WRITE A PROGRAM THAT WILL READ FROM THE TERMINAL THE FOLLOWING:

TYPE OF LIQUOR
MANUFACTURER
QUANTITY
UNIT PRICE

AND STORE IN A DATA FILE THE:

TYPE OF LIQUOR
MANUFACTURER
QUANTITY
UNIT PRICE
INVENTORY VALUE (QUANTITY * UNIT PRICE)

LEGAL TYPES OF LIQUORS ARE SCOTCH, RUM, GIN, VODKA, AND RYE. LEGAL MANUFACTURERS ARE SEAGRAMS, JOHNNY WALKER, BACARDI, BEEFEATERS, SMIRNOFF, ABSOLUT, DEWARS, AND GORDONS. MOST OF THE TIME (OVER 50%) THE TYPE OF LIQUOR IS GIN AND THE MANUFACTURER IS BEEFEATERS.

HINT: USE RECORD TYPES, POSSIBLY WITH DEFAULTS.

B. ADD A SUBPROGRAM WHICH COMPUTES THE TOTAL VALUE OF THE STOCK ROOM.

HINT: USE THE DATA FILE CREATED IN PART A.

INSTRUCTOR NOTES

ASSIGN AFTER L202 SECTION 9 HAS BEEN TAUGHT.

GIVE THEM THE NAME OF THE TEST DATA FILE.

i) PROVIDES A GOOD USE FOR A RECORD WITH A DISCRIMINANT.

EXERCISE 22

IN THIS EXERCISE WE WILL BEGIN DEVELOPMENT OF A VERY MUCH SIMPLIFIED "AIR TRAFFIC CONTROLLER" PACKAGE. WE WILL CONCENTRATE ON DEFINING THE DATA TYPES NEEDED TO REPRESENT INFORMATION ABOUT AN AIRCRAFT IN FLIGHT, AND THE DATA STRUCTURES NEEDED BY THE AIR TRAFFIC CONTROL STATION TO KEEP TRACK OF THE PLANES IN ITS AREA.

THE FOLLOWING INFORMATION IS NEEDED TO DESCRIBE AN AIRCRAFT IN FLIGHT. WRITE DEFINITIONS FOR THE DATA TYPES REQUIRED TO REPRESENT EACH ITEM INCLUDING ANY SUPPORTING TYPES REQUIRED. UNLESS DIRECTED OTHERWISE, USE YOUR OWN JUDGMENT ABOUT THE SPECIFIC CHARACTERISTICS OF THE ITEM.

a) IDENTIFICATION CODE -- A UNIQUE IDENTIFIER FOR THE PARTICULAR AIRCRAFT

b) AIRLINE -- CAN BE ONE OF AMERICAN, TWA, EASTERN, UNITED, U.S. AIR, NEW YORK AIR, REPUBLIC

c) FLIGHT NUMBER

d) POSITION -- CONSISTS OF LATITUDE AND LONGITUDE

e) ATTITUDE -- THE DIRECTION THE PLANE IS POINTING: CONSISTS OF THREE ANGLES CALLED ROLL, PITCH, AND YAW

f) ALTITUDE

g) VELOCITY -- BOTH AIR SPEED AND GROUND SPEED SHOULD BE RECORDED

h) TYPE OF AIRCRAFT -- CAN BE ONE OF L1011, B727, B747, DC10, B767, AIRBUS, DC9, DE HAVILLAND OTTER

i) FLIGHT PLAN -- CONSISTS OF ONE CITY OF ORIGIN PLUS 1-5 DESTINATION CITIES

j) NUMBER OF PASSENGERS

k) DEPARTURE TIME -- DATE AND TIME OF DEPARTURE FROM CITY OF ORIGIN

l) CONTROLLING CITY -- THE CITY WHOSE AIR TRAFFIC CONTROL CENTER IS RESPONSIBLE FOR THE AIRCRAFT -- MAY BE ONE OF BOSTON, WASHINGTON, ATLANTA, CHICAGO, DENVER, DALLAS, LOS ANGELES, ALBUQUERQUE

INSTRUCTOR NOTES

A. DEFINE A RECORD TYPE THAT CAN BE USED TO DESCRIBE COMPLETELY AN AIRCRAFT IN FLIGHT. IT SHOULD INCLUDE ALL OF THE CHARACTERISTICS OF THE DEFINED TYPES .

B. A PARTICULAR CONTROL CENTER MUST KEEP TRACK OF ALL AIRCRAFT WITHIN ITS AREA, ADDING AND DELETING THEM AS REQUIRED. WRITE A DECLARATION FOR A DATA STRUCTURE THAT CAN BE USED TO STORE THE DATA ON ALL AIRCRAFT IN THE AREA. ALSO, DECLARE A DATA ITEM THAT WILL CONTAIN THE COUNT OF AIRCRAFT IN THE AREA. YOU WILL, FOR LATER EXERCISES, NEED TO BE ABLE TO ADD AND DELETE ENTRIES OF THIS DATA STRUCTURE.

INSTRUCTOR NOTES

C. WRITE A SUBPROGRAM Find_Plane THAT, GIVEN AN AIRCRAFT'S IDENTIFICATION CODE, RETURNS THE RECORD DESCRIBING IT. Find_Plane SHOULD RETURN A RECORD AND A POSITION IN AN ARRAY. HANDLE THE CASE WHERE IT IS NOT IN THE AIRCRAFT DATA STRUCTURE IN SOME APPROPRIATE MANNER.

D. WRITE A SUBPROGRAM Add_Plane THAT, GIVEN A RECORD DESCRIBING AN AIRCRAFT, ADDS IT TO THE AIRCRAFT DATA STRUCTURE. HANDLE IN SOME APPROPRIATE MANNER THE CASE WHERE THE DATA STRUCTURE IS ALREADY FULL.

INSTRUCTOR NOTES

E. WRITE A FUNCTION Number_Of_Planes THAT RETURNS THE NUMBER OF PLANES IN THE AIRCRAFT DATA STRUCTURE.

F. WRITE A FUNCTION Number_Of_Planes_At_Altitude THAT, GIVEN AN ARGUMENT THAT IS AN ALTITUDE, RETURNS THE NUMBER OF PLANES AT THAT ALTITUDE $\pm$ A PREDEFINED DELTA (OF YOUR CHOOSING).

INSTRUCTOR NOTES

ASSIGN AFTER L202 SECTION 10 HAS BEEN TAUGHT.

EXERCISE 23

MODIFY THE DATA STRUCTURE STORING THE AIRCRAFT DATA IN EXERCISE 24 SO THAT IT CONTAINS POINTERS TO AIRCRAFT RECORDS RATHER THAN HOLDING THE ACTUAL RECORDS.

INSTRUCTOR NOTE

ASSIGN AFTER L202 SECTION 11 HAS BEEN TAUGHT.

EXERCISE 24

ADD A SUBPROGRAM Delete_Plane TO THE PROGRAM FROM EXERCISE 22 THAT, GIVEN AN AIRCRAFT IDENTIFICATION CODE, DELETES IT FROM THE AIRCRAFT DATA STRUCTURE. (YOU WILL NEED TO DECIDE ON SOME CONVENTION FOR INDICATING "DELETED" ENTRIES.)

HINT: USE Find_Plane.

INSTRUCTOR NOTES

ASSIGN AFTER L202 SECTION 11 HAS BEEN TAUGHT.

EXERCISE 25

ADD A SUBPROGRAM Process_New_Plane_Input THAT, GIVEN A RECORD DESCRIBING AN AIRCRAFT, DOES THE FOLLOWING:

1) IF THE NEW PLANE'S CONTROLLING CITY IS ALBUQUERQUE AND THE PLANE IS ALREADY IN THE AIRCRAFT DATA STRUCTURE, REPLACE THE OLD INFORMATION WITH THAT IN THE NEW RECORD.

2) IF THE NEW PLANE'S CONTROLLING CITY IS ALBUQUERQUE AND IT IS NOT IN THE DATA STRUCTURE, ADD IT.

3) IF THE NEW PLANE'S CONTROLLING CITY IS NOT ALBUQUERQUE BUT IT IS IN THE DATA STRUCTURE (I.E., IT HAS MOVED TO A NEW CONTROL AREA), DELETE IT.

INSTRUCTOR NOTES

ASSIGN AFTER L202 SECTION 12 HAS BEEN TAUGHT.

POINT OUT THAT BOTH A AND B START BY MODIFYING THE SAME PROGRAM. THE ASSIGNMENT IS TO MODIFY IT TWO DIFFERENT WAYS.

EXERCISE 26

A. MODIFY THE PROGRAM FROM EXERCISE 16 SO THAT THE TYPES AND PROCEDURES ARE IN A PACKAGE WHICH GETS "WITH"ED BY THE MAIN PROCEDURE.

B. MODIFY THE PROGRAM FROM EXERCISE 16 SO THAT THE PROCEDURES AND FUNCTIONS ARE STUBBED OUT OF THE MAIN PROCEDURE.

INSTRUCTOR NOTES

ASSIGN AFTER L202 SECTION 13 HAS BEEN TAUGHT.

EXERCISE 27

CREATE A PACKAGE Air_Traffic_Controller TO HOLD THE TYPE DEFINITIONS, DATA STRUCTURES AND SUBPROGRAMS DESIGNED IN EXERCISES 22, 25, AND 27. ALL SUBPROGRAMS SHOULD BE VISIBLE TO THE USER OF THE PACKAGE. THE ACTUAL NUMBER OF PLANES SHOULD NOT BE ACCESSIBLE BY THE USER; FUNCTION Number_of_Planes PROVIDES THIS INFORMATION.

INSTRUCTOR NOTES

ASSIGN AFTER SECTION 14 OF L202 IS TAUGHT.

EXERCISE 28

MODIFY THE PROGRAM FROM EXERCISE 5 SO THAT THE PROGRAM TERMINATES GRACEFULLY WHEN AN UNRECOGNIZABLE LANGUAGE IS ENTERED.

HINT: USE EXCEPTION HANDLERS.

INSTRUCTOR NOTES

ASSIGN AFTER SECTION 14 OF L202 IS TAUGHT.

EXERCISE 29

MODIFY THE PROGRAM FROM EXERCISE 4 SO THAT THE PROGRAM TERMINATES GRACEFULLY WHEN AN UNACCEPTABLE CHARACTER IS ENTERED.

HINT: USE EXCEPTION HANDLERS.

INSTRUCTOR NOTES

ASSIGN AFTER L202 SECTION 14 HAS BEEN TAUGHT.

EXERCISE 30

A. MODIFY THE PROGRAM FROM EXERCISE 22D TO RAISE AN EXCEPTION Too_Many_Planes IF ASKED TO ADD AN AIRCRAFT WHEN ITS DATA STRUCTURE IS ALREADY FULL.

B. MODIFY THAT SAME PROGRAM TO RAISE AN EXCEPTION Plane_Too_Low IF ASKED TO ADD AN AIRCRAFT WHOSE ALTITUDE IS BELOW A PREDEFINED LIMIT (OF YOUR CHOOSING). IN THIS CASE, THE AIRCRAFT SHOULD NOT BE ADDED TO THE DATA STRUCTURE.

C. MODIFY IT TO RAISE AN EXCEPTION Too_Many_Planes_At_Same_Altitude IF ASKED TO ADD AN AIRCRAFT WHEN MORE THAN SOME PREDEFINED NUMBER (OF YOUR CHOOSING) OF PLANES IN THE AREA ARE ALREADY WITHIN THAT ALTITUDE $\pm$ A PREDEFINED DELTA. IN THIS CASE, THE AIRCRAFT SHOULD NOT BE ADDED.

INSTRUCTOR NOTES

ASSIGN AFTER L202 SECTION 14 HAS BEEN TAUGHT.

EXERCISE 31

MODIFY THE PROGRAM FROM EXERCISE 9 SO THAT EXECUTION STOPS WHEN A NON-THREE-DIGIT NUMBER IS ENTERED.

HINT: USE EXCEPTION HANDLERS.

INSTRUCTOR NOTES

ASSIGN AFTER SECTION 15 OF L202 HAS BEEN TAUGHT.

EXERCISE 32

WRITE A SUMMARY REPORT FOR THE ELOD COMPANY. AN EXAMPLE FORMAT IS SHOWN ON THE FOLLOWING PAGE.

HINT: USE THE PROCEDURES AND FUNCTIONS DEFINED IN EXERCISE 26.

INSTRUCTOR NOTES

ELOD ANNUAL REPORT

TREE NUMBER	1	2	3	4	5	6	7	8	9	10	TOTAL
HAWAII											
GRAPEFRUIT	X	X	X	X	X	X	X	X	X	X	XX
PINEAPPLE	X	X	X	X	X	X	X	X	X	X	XX
LIMES	X	X	X	X	X	X	X	X	X	X	XX
TOTAL	XX	XX	XX	XX	XX	XX	XX	XX	XX	XX	XXX
MEXICO											
GRAPEFRUIT	X	X	X	X	X	X	X	X	X	X	XX
PINEAPPLE	X	X	X	X	X	X	X	X	X	X	XX
LIMES	X	X	X	X	X	X	X	X	X	X	XX
TOTAL	XX	XX	XX	XX	XX	XX	XX	XX	XX	XX	XXX
VIRGIN ISLANDS											
GRAPEFRUIT	X	X	X	X	X	X	X	X	X	X	XX
PINEAPPLE	X	X	X	X	X	X	X	X	X	X	XX
LIMES	X	X	X	X	X	X	X	X	X	X	XX
TOTAL	XX	XX	XX	XX	XX	XX	XX	XX	XX	XX	XX
GRANDTOTAL	XX	XX	XX	XX	XX	XX	XX	XX	XX	XX	XXX

END

FILMED

5-86

DTIC